PRACTICA CALIFICADA 03

Asignatura: Desarrollo de Aplicaciones Web
Docente: Mg. Jaime Suasnabar Terrel
Nombre del Estudiante: De la Cruz Salazar Franklin Efraín

1. Construir una aplicación web que muestre un reloj analógico utilizando html canvas, css y javascript.

Código:

reloj.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reloj con Canvas</title>
    <link rel="stylesheet" href="reloj.css">
</head>
<body>
    <H1>RELOJ</H1>
    <canvas id="canvas" width="400" height="400"></canvas>
    <script src="reloj.js"></script>
</body>
</html>

```

reloj.css

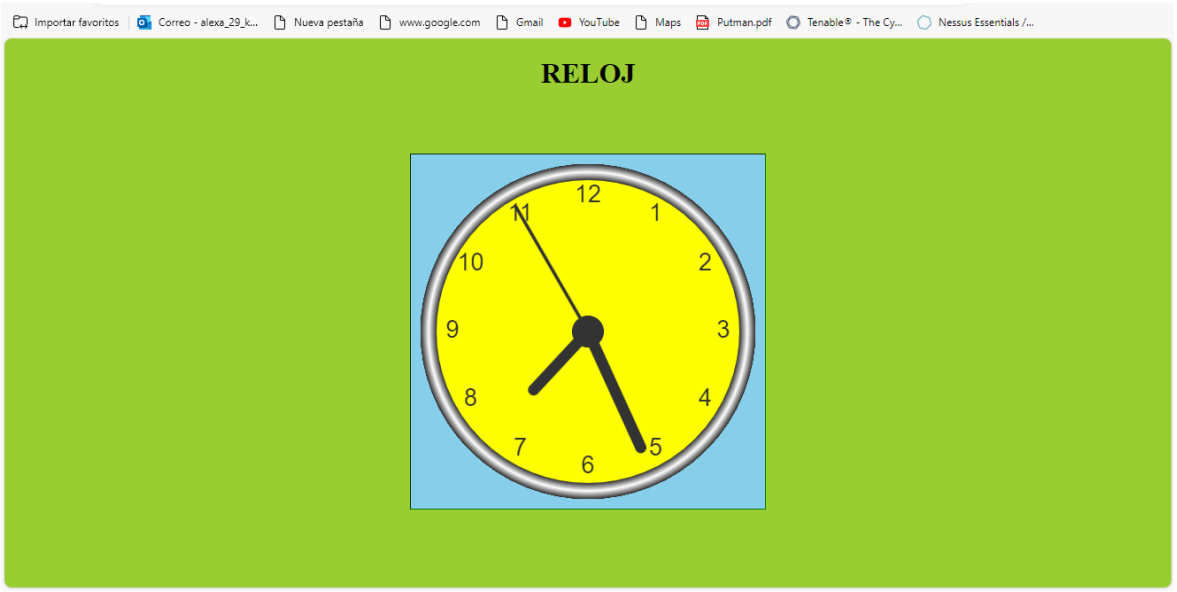
```css
body{
    background: yellowgreen;
    text-align: center;
}

#canvas{
    border:thin inset green;
    background: skyblue;
    margin-top: 50px;

    display: inline-block;
}
```

JS reloj.js > ...

```
var canvas = document.getElementById("canvas");
var context = canvas.getContext("2d");
var radius = canvas.height / 2;
context.translate(radius, radius);
radius = radius * 0.90


setInterval(drawClock,1000);

function drawClock() {
    drawFace(context,radius);
    drawNumbers(context,radius);
    drawTime(context,radius);
}

function drawFace(ctx, radius) {
    var grad;

    context.beginPath();
    context.arc(0, 0, radius, 0, 2*Math.PI);
    context.fillStyle = 'yellow';
    context.fill();

    grad = context.createRadialGradient(0,0,radius*0.95, 0,0,radius*1.05);
    grad.addColorStop(0, '#333');
    grad.addColorStop(0.5, 'white');
    grad.addColorStop(1, '#333');

    context.strokeStyle = grad;
    context.lineWidth = radius*0.1;
    context.stroke();
```

```

function drawTime(ctx, radius){

    var now = new Date();
    var hour = now.getHours();
    var minute = now.getMinutes();
    var second = now.getSeconds();

    hour=hour%12;
    hour=(hour*Math.PI/6)+(minute*Math.PI/(6*60))+(second*Math.PI/(360*60));
    drawHand(ctx, hour, radius*0.5, radius*0.07);

    minute=(minute*Math.PI/30)+(second*Math.PI/(30*60));
    drawHand(ctx, minute, radius*0.8, radius*0.07);

    second=(second*Math.PI/30);
    drawHand(ctx, second, radius*0.9, radius*0.02);
}

function drawHand(ctx, pos, length, width) {
    ctx.beginPath();
    ctx.lineWidth = width;
    ctx.lineCap = "round";
    ctx.moveTo(0,0);
    ctx.rotate(pos);
    ctx.lineTo(0, -length);
    ctx.stroke();
    ctx.rotate(-pos);
}
```

Muestra del Código:



2. Construir una aplicación web que muestre un gráfico pastel con áreas de 10%, 20%, 30% y 40% con 4 colores utilizando html canvas, css y javascript.

Código:

pastel.html > html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Gráfico de Pastel</title>
    <link rel="stylesheet" href="pastel.css">
</head>
<body>
    <H1>GRÁFICO DE PASTEL</H1>
    <div class="grafico">
        <canvas id="graficoCanvas" width="400" height="400"></canvas>
    </div>
    <script src="pastel.js"></script>
</body>
</html>
```

pastel.css > ...

```css

body {
    font-family: Arial, sans-serif;
    text-align: center;
    background-color: skyblue;
}

.grafico {
    border: 4px solid black;
    display: inline-block;
    margin-top: 50px;
    background: blanchedalmond;
}
```

JS pastel.js > ...

```
var canvas = document.getElementById('graficoCanvas');
    context = canvas.getContext('2d');
    data = [10, 20, 30, 40];
    colors = ['red', 'blue', 'green', 'orange'];
    centerX = canvas.width / 2;
    centerY = canvas.height / 2;
    radius = Math.min(centerX, centerY) - 10;

function dibujarGrafico() {
    let startAngle = 0;
    for (let i = 0; i < data.length; i++) {
            sliceAngle = (data[i] / 100) * 2 * Math.PI;

        context.beginPath();
        context.moveTo(centerX, centerY);
        context.arc(centerX, centerY, radius, startAngle, startAngle + sliceAngle);
        context.closePath();
        context.fillStyle = colors[i];
        context.fill();


          labelX = centerX + (radius / 2) * Math.cos(startAngle + sliceAngle / 2);
          labelY = centerY + (radius / 2) * Math.sin(startAngle + sliceAngle / 2);
        context.fillStyle = 'black';
        context.font = '24px Arial';
        context.textAlign = 'center';
        context.fillText(data[i] + '%', labelX, labelY);

        startAngle += sliceAngle;
    }
}
dibujarGrafico();
```

Muestra del Código: